IN THE CLAIMS

Please amend the claims as follows:

1. (Previously Presented) A system comprising:

a storage element;

a memory hierarchy coupled to the storage element; and

a processor coupled to the memory hierarchy, wherein in response to receiving a single replay indicator, the processor to repeatedly execute a plurality of instructions a plurality of times using a replay handler loaded into the memory hierarchy, wherein the processor does not require receipt of another replay indicator to repeatedly execute the plurality of instructions, and wherein the processor is to compare results from one of the executions of the plurality of instructions to results from another of the executions of the plurality of instructions.

2. (Original) The system of claim 1 wherein the memory hierarchy is an instruction cache.

3. (Canceled)

4. (Previously Presented) The system of claim 1 wherein the replay handler includes the plurality of instructions.

5. (Previously Presented) The system of claim 1 wherein the replay handler loads the plurality of instructions into the memory hierarchy from an external device.

6 - 9. (Canceled)

10. (Previously Presented) A system comprising:

a memory hierarchy;

a processor coupled to the memory hierarchy wherein the processor is to execute instructions from the memory hierarchy;

a port coupled to the processor and memory hierarchy; and

a host system coupled to the port, the host system to generate a replay handler, to generate at least one execution to be repeatedly replayed by the processor when executing the replay handler, and to generate a signal to the processor to cause the processor to load the replay handler into the memory hierarchy and to cause the processor to repeatedly replay the at least one execution, wherein the processor is capable of repeatedly replaying the at least one execution without generation of another signal to cause the processor to load the replay handler into the memory hierarchy, and wherein the host system is to compare a first result of replaying the at least one execution to a second result of replaying the at least one execution.

11. (Previously Presented) The system of claim 10, wherein after the signal is generated, the processor to save original code of the memory hierarchy, the processor to load the replay handler into the memory hierarchy from the host system through the port, and the processor to execute the replay handler.

12. (Previously Presented) The system of claim 11, wherein after the replay handler is executed, the host system to modify the replay handler.

13. (Original) The system of claim 12 wherein the replay handler is modified to alter starting and stopping points of one of the at least one executions.

14. (Original) The system of claim 10 wherein a replay state is sent to the host system through the port.

15. (Original) The system of claim 10, wherein the port is a network interface.

16. (Original) The system of claim 10, wherein the port is a serial interface.

17 – 19. (Canceled)

20. (Previously Presented) A method comprising:

testing a processor in response to a replay break including,

interrupting processes executing on the processor;

storing minimal state information sufficient to later resume the interrupted processes;

storing original code of an instruction cache;

loading a replay handler into the instruction cache;

branching execution of the processor to the replay handler;

replaying a system execution a number of times from a starting point to a stopping point while monitoring state information to test for proper operation of the processor, wherein the replaying the system execution a number of times does not require responding to another replay break;

comparing results of replaying the execution one time to results of replaying the execution other times.

loading the original code into the instruction cache; and

resuming interrupted processes utilizing the minimal state information.

21. (Previously Presented) The method of claim 20,wherein the number of times, the starting point, and the stopping point were modified by a user.

22. (Original) The method of claim 20 further comprising:

generating the system execution by tracing an execution of a program.

23. (Previously Presented) A computer readable medium containing computer instructions for instructing a processor to perform a method of:

generating at least one execution that includes a plurality of processor instructions; and

testing the processor in response to the processor receiving a single replay break, wherein the testing includes,

interrupting normal processing;

loading a replay handler into a memory hierarchy;

repeatedly replaying the at least one execution to test for proper operation of the processor, wherein the repeatedly replaying the at least one execution does not require the receipt of another replay break;

comparing results of replaying the execution one time to results of replaying the execution multiple times;

accessing state information;

storing state information; and

resuming normal processing.

24. (Previously Presented) The computer readable medium of claim 23, wherein the replay handler includes computer instructions which, when executed, cause the repeatedly replaying the at least one execution to occur.

25. (Previously Presented) The computer readable medium of claim 23, wherein the replay handler has a predetermined number of replays for the at least one execution.

26. (Previously Presented) The computer readable medium of claim 23, wherein the replay handler dynamically determines the number of replays for the at least one execution.